

Basic Scheduling Problems with Raw Material Constraints

Alexander Grigoriev,¹ Martijn Holthuijsen,² Joris van de Klundert²

¹ Faculty of Economics and Business Administration, University of Maastricht, FdEWB/KE,
P.O. Box 616, Maastricht 6200 MD, The Netherlands

² Department of Mathematics, University of Maastricht, P.O. Box 616, Maastricht 6200 MD, The Netherlands

Received 18 February 2004; revised 23 March 2005; accepted 13 April 2005

DOI 10.1002/nav.20095

Published online 16 June 2005 in Wiley InterScience (www.interscience.wiley.com).

Abstract: One of the achievements of scheduling theory is its contribution to practical applications in industrial settings. In particular, taking finiteness of the available production capacity explicitly into account, has been a major improvement of standard practice. Availability of raw materials, however, which is another important constraint in practice, has been largely disregarded in scheduling theory. This paper considers basic models for scheduling problems in contemporary manufacturing settings where raw material availability is of critical importance. We explore single scheduling machine problems, mostly with unit or all equal processing times, and L_{\max} and C_{\max} objectives. We present polynomial time algorithms, complexity and approximation results, and computational experiments. © 2005 Wiley Periodicals, Inc. *Naval Research Logistics* 52: 527–535, 2005.

Keywords: scheduling; raw material requirements; computational complexity; approximation algorithms

1. INTRODUCTION

This paper deals with scheduling problems that arise in the context of producing to order. The presented models and problems take the availability of raw materials explicitly into account. Such scheduling problems, in which raw material availability and production capacity are jointly considered, arise naturally in state of the art manufacturing settings. To the best of our knowledge, however, they have been almost completely unexplored in the scheduling literature. We now first describe the practical context more precisely.

A manufacturing strategy that strives to combine rapid fulfillment with customized products, is the Assemble To Order strategy. In such a strategy, which is now common in most of the important industries such as high tech and automotive, the last production stage, called assembly, is to customer order, whereas the preceding stages are to stock. The product is assembled to be customer specific using standard components, and therefore the assembly stage serves as the customer order decoupling point. This means that there are a limited number of raw materials from which a multitude of different end products can be delivered to the

customer. In such environments it is important to deliver quickly, but at low cost, and therefore with limited raw material inventory.

While scheduling assembly orders in such an Assemble To Order environment, one has to cope with tight due dates and limited raw material availability. MRP (Materials Requirement Planning; see, e.g., [1]) systems are known to fall short in such a setting, since they cannot cope with the finite capacities of production resources. Moreover, MRP systems cannot react quickly to short-term changes which are common in practice: customer orders are canceled, due dates are changed, rush orders are accepted, deliveries are late, deliveries are in smaller quantities, etc. Scheduling theory, on the other hand, which is in principle well suited to deal with the details of short term problems, has until now by and large ignored raw material availability. As a result, while practice is working to solve these problems, very little theory has been developed. In this paper, we therefore investigate several of the most basic problems, and identify a number of related problems for further research.

Before proceeding, we briefly overview related little literature on scheduling problems. There is, of course, much literature on scheduling problems with release dates (see, e.g., [4]), and indeed release dates capture some of the issues dealt with in the aforementioned problem setting. Release dates model the fact that a production order is not

Correspondence to: A. Grigoriev (a.grigoriev@ke.unimaas.nl)

allowed to start before a certain moment in time, its release date. In Make To Order environments, it can be customary to order all raw materials per customer order. Hence, for each of the production stages, the release date can be set to be the earliest moment at which all the raw materials have arrived. Release dates, however, are insufficient in settings where the ordering of raw materials is decoupled from the customer orders, as, for instance, in Assemble To Order environments.

Alternatively, there are results on scheduling problems with renewable resources (see, for instance, [3]), but, as the name already suggests, they appear to be more fitting for resource constrained problems than for raw materials constrained problems.

In research on planning and scheduling problems in the process industry, where mixing of raw materials is more critical, raw materials availability is often incorporated into the models (see, for instance, [9]). However, the models considered are often rather complex, for instance, since they incorporate nonlinear relationships to model the chemical processes. In addition, process industry usually produces to stock, and altogether the nature of these problems is quite different from the assemble to order problems we focus on. Nevertheless, standard software has become available in which it is possible to model raw materials availability. In ILOG Scheduler, for instance, they can be modeled using “reservoirs” (see [11]).

This paper considers three models for raw materials usage. First of all, it may be the case that each customer order requires its own unique raw material type. Then, there is no assignment decision to be taken as to which quantity of raw materials is to be used for a customer order. As explained above, such problems can be satisfactorily modeled using release dates, and hence we will only briefly review them in subsequent sections.

A second simple model for raw materials usage arises when all customer orders require a single common raw material. This setting may appear somewhat theoretical, but in practical production environments it may be the case that for only one of several raw materials availability is critical. For theoretical reasons the case is interesting, because it is a basic special case. The cases we explored can be solved in polynomial time. In fact, it will appear that the optimal sequence of orders does not depend on the arrival of raw materials.

The third model we consider, is the case where there are many raw materials which may be required for each of the customer orders. The cases we explored appear to be NP-hard, and we investigate approximation properties, and perform some computational experiments.

We mostly study the case in which there is only one machine, and all jobs have unit or all equal processing times. This model is simple, yet it essentially captures the

behavior of an assembly line that produces items at a constant rate, as commonly encountered in the Assemble To Order environment that motivates this paper [10, 13]. Since the problem deals with the issue of satisfying customer demand in time, an interesting question to ask is whether it is at all possible to satisfy all customer orders in time, given a schedule of raw material arrivals. This feasibility problem can be translated to an optimization setting in various ways. We have chosen to study the L_{\max} criterion. Indeed, then the feasibility problem translates to the question whether there is a solution with $L_{\max} = 0$. Moreover, in case the feasibility problem cannot be solved, the L_{\max} criterion appears reasonable. Further, it has been widely investigated in the scheduling literature in problems with release dates.

Besides the L_{\max} objective function, the paper contains results on problems with a C_{\max} objective function. It may serve as a reasonable model in some cases, for instance when minimizing operating hours, but in any case it is a fundamental objective function.

This paper is organized as follows. Section 2 clarifies the mathematical models and the notations. Sections 3 and 4 investigate the complexity of the problems with L_{\max} and C_{\max} objectives, respectively, and contain worst case analysis of polynomial time approximation algorithms as well. Section 5 discusses computational results on problems with L_{\max} and C_{\max} objective functions. Section 6 contains a discussion and directions for further research.

2. MODELING AND NOTATIONS

In this paper it is convenient to adopt the three-field scheduling notation $\alpha|\beta|\gamma$ described in [6]. The problems under investigation are related to $1|p_j = 1|L_{\max}, C_{\max}$ and $1|p_j = p|L_{\max}, C_{\max}$. We shall extend the notation by introducing a parameter for raw materials. We present two additional scripts *ddc* and *rm* where the script *ddc* corresponds to different dedicated raw material for each job, and *rm* = *m* means we have *m* raw materials. The parameter value *rm* denotes the case where raw materials are not dedicated, and the number of raw materials is part of the input. Throughout, we assume all numbers to be integral.

We will refer to the customer orders as jobs. Let $J = \{1, \dots, n\}$ be the set of jobs. Each job may have a due date d_j , $j \in J$. Moreover, each job requires raw materials. In general, let $I = \{1, \dots, m\}$ be the set of raw materials. If there is only one raw material, let a_j be the amount of the raw material required by job $j \in J$. If there are several different raw materials, a_j may be viewed as a vector, where $a_{i,j}$ denotes the requirement of raw material $i \in I$ by job $j \in J$. We denote by $r_{i,t}$ the amount of raw material i delivered at time t , and let $R_i(t) = \sum_{\tau=0}^t r_{i,\tau}$. We omit the index i in the case where there is only one raw material. Through the paper we assume that within a finite time horizon $[0, T]$,

the supply of raw materials is exactly sufficient to complete all jobs, i.e.,

$$R_i(T) \geq \sum_{j=1}^n a_{ij}, \quad \forall i \in I. \quad (1)$$

This requirement can be easily verified in $O(mn + mT)$ time and, by default, we assume that all algorithms in this paper perform this verification in the preprocessing phase.

In an arbitrary schedule S , we let s_j denote the starting time of job j . Then, for schedule S to be feasible, it must hold that

$$\sum_{j: s_j \leq t} a_{ij} \leq R_i(t), \quad \forall i \in I, \quad \forall t \in \{0, \dots, T\}, \quad (2)$$

$$\sum_{j: t_1 \leq s_j \leq t_2} p \leq \left\lfloor \frac{t_2 - t_1}{p} \right\rfloor + 1, \quad \forall t_1 \in \{0, \dots, t_2\}, \quad \forall t_2 \in \{0, \dots, T\}, \quad (3)$$

where p is the common processing time of all jobs.

The first inequality models that the raw materials consumed until time t cannot exceed the amount delivered until t . The second inequality models that in any period of length $t_2 - t_1$ on the single nonsharable machine at most $\lfloor (t_2 - t_1)/p \rfloor + 1$ jobs with processing time p can be started. In the remainder we say schedule S is feasible if and only if it satisfies inequalities (2) and (3).

The completion time C_j of job j is defined as $C_j = s_j + p$, and the lateness L_j of job j is defined as $L_j = C_j - d_j$. Throughout the paper we assume that the jobs are ordered in such a way that $d_1 \leq d_2 \leq \dots \leq d_n$. Thus, jobs are in nondecreasing order of their due dates. This order is known as the Earliest Due Date (EDD) order. Further, in the case where there is only one raw material, i.e., $rm = 1$, we assume that if $d_j = d_{j+1}$, then $a_j \leq a_{j+1}$. In words, jobs having the same due date are ordered in nondecreasing order of their raw material requirement.

3. MINIMIZATION OF MAXIMUM LATENESS

This section deals with three problems, namely $1|ddc, \cdot |L_{\max}, 1|rm = 1, p_j = 1|L_{\max}, 1|rm, p_j = p|L_{\max}$. For any moment in time t , we let $J(t)$ be the set of jobs j for which $d_j \leq t$.

3.1. Dedicated Raw Materials ($1|ddc, \cdot |L_{\max}$)

As already briefly mentioned, problems in which each job has its own dedicated raw materials, can be adequately

modeled by the well known concept of release dates. Hence, $1|ddc, \cdot | \cdot$ problems are equivalent to $1|r_j, \cdot | \cdot$ problems studied in the classical scheduling literature (see, e.g., [6]). It is well known that the problem $1|r_j|L_{\max}$ is strongly NP-Hard (see [12]). Polynomial time algorithms are known for problems $1|r_j, prec, p_j = p|L_{\max}$ and $1|r_j, prec, pmtn|L_{\max}$ (see [14] and [2], respectively).

3.2. One Raw Material, Unit Processing Times ($1|rm = 1, p_j = 1|L_{\max}$)

We define

$$A(t) = \sum_{j \in J(t)} a_j. \quad (4)$$

Let $u_1 \leq u_2 \leq \dots \leq u_q$ be the moments in time for which $r_t \geq 1$, i.e., the time moments when a positive amount of the single raw material is supplied. Since we assume that the sum of the delivered quantities is exactly sufficient to complete all jobs, u_q is the earliest moment in time at which all required raw materials have arrived. Let $U = \{u_1, u_2, \dots, u_q\} \cup \{d_1, d_2, \dots, d_n\}$. Thus, U is the set of time moments at which either a (nonzero) quantity of the raw material arrives, or some job has its due date.

Notice that for all $u \in U$ quantities $R(u)$ and $A(u)$ can be straightforwardly computed in times $O(T)$ and $O(nT)$ respectively.

THEOREM 1: $1|rm = 1, p_j = 1|L_{\max}$ can be solved in time $O(n^2)$.

PROOF: Let us first consider the case where all due dates are distinct. Consider a schedule S where each job is scheduled such that it is completed at its due date: $s_j = d_j - 1$. It follows from the unit processing times and unique due dates that there is sufficient time to process all jobs in time in S , and hence that inequality (3) is satisfied. Hence, the feasibility of the schedule depends on the raw materials availability, as modeled by inequality (2). Indeed, schedule S is feasible if and only if $R(t) \geq A(t)$, $\forall 0 \leq t \leq d_n$. Once $R(U)$ and $A(U)$ are computed, it can be easily checked in time linear in $q + n$ whether $R(u) \geq A(u)$, $\forall u \in U$, and hence, whether a schedule with zero lateness exists.

To check whether a schedule with lateness L exists is now also easily done. Simply add L to all due dates, and check whether a zero lateness schedule exists. Hence, we have derived that the EDD order minimizes the maximum lateness in the case where all due dates are distinct. Moreover, the EDD order is optimal regardless of the $A(t)$, $0 \leq t \leq T$. Hence, whenever jobs have distinct due dates, the optimal solution can be found in $O(n \log n)$ times, required to sort them on EDD order.

Now suppose that the job due dates are not unique, i.e. several jobs may share a common due date. In that case, inequality (3) is not so easily satisfied. The following observations provide the solution to this issue.

OBSERVATION 2: Let l , $1 \leq l \leq n - 1$, be such that $d_l = d_{l+1}$, and hence $a_l \leq a_{l+1}$. Let S and S' be schedules such that

$$s_l = s'_{l+1}, \quad (5)$$

$$s_{l+1} = s'_l, \quad (6)$$

$$s_j = s'_j, \quad \forall j \notin \{l, l+1\}. \quad (7)$$

Further, let $s_{l+1} < s_l$. Then S' is feasible if S is feasible.

By consequence, we may assume without loss of generality that in an optimal solution, the jobs with equal due dates are in order of non decreasing raw materials requirements. This leads us to the following observation:

OBSERVATION 3: Let I be an instance of $1|rm = 1$, $p_j = 1|L_{\max}$, where index l is any index such that $d_l = d_{l+1}$. Construct an instance I' which results from setting $d'_l = d_l - 1$, and copying all other parameters from I . Then any solution that is optimal for I' is optimal for I as well. Moreover, I' and I have the same optimal solution value.

From Observations 2 and 3, one easily verifies the correctness of the following polynomial algorithm to convert an arbitrary instance of $1|rm = 1$, $p_j = 1|L_{\max}$, to an instance with the same solution value, where all jobs have unique due dates:

1. Set $j = n - 1$. While $j > 1$ do:
2. If $d_j = d_{j+1}$, set $d_j = d_j - 1$, and reinsert job j such that $d_1 \leq d_2 \leq \dots \leq d_n$ and jobs sharing a common due date are indexed in nondecreasing order of their raw material consumption.
3. If $d_j \neq d_{j+1}$, set $j := j - 1$.

A more efficient implementation of the final algorithm would be the following backward scheduling rule where the jobs become available at their due dates: *At any time schedule an available job j with the largest raw material consumption a_j .* This rule can be implemented in $O(n \log n)$ (see [7]). \square

In the analysis above, we already observed that, once the due dates are distinct, the same EDD sequence of jobs is optimal, regardless of the arrivals of raw materials. Moreover, since any instance can be transformed into an instance with distinct due dates, it is possible to derive an optimal

order for every instance, which is optimal regardless of the raw materials arrivals. This implies that the problem can be solved to optimality once all jobs are known, even when the $r(t)$ are revealed online. In other words, it is 1-competitive.

3.3. Two Raw Materials, All Equal Processing Times ($1|rm = 2, p_j = p|L_{\max}$)

In this section we first show that even $1|rm = 2, p_j = 1|L_{\max}$ is strongly NP-hard.

THEOREM 4: The problem $1|rm = 2, p_j = 1|L_{\max}$ is strongly NP-hard.

PROOF: Let us reduce the well known strongly NP-hard problem 3-PARTITION, see [5], to $1|rm = 2, p_j = 1|L_{\max}$.

3-PARTITION: Given a set E of $3m$ elements, a bound $B \in \mathbb{Z}^+$, and size $s(e) \in \mathbb{Z}^+$ for each $e \in E$ such that $B/4 < s(e) < B/2$ and $\sum_{e \in E} s(e) = mB$, can E be partitioned into m disjoint sets E_1, E_2, \dots, E_m such that for $1 \leq k \leq m$, $\sum_{e \in E_k} s(e) = B$?

Using E to refer to the set of jobs, we construct the reduction as follows. Define $J = E$,

$$\begin{aligned} d_e &= 3m, \quad a_{1,e} = s(e), \quad e \in E, \\ a_{2,e} &= B - s(e), \quad e \in E, \\ r_{1,t} &= B \quad \text{if } t \bmod 3 = 0 \quad \text{and} \quad r_{1,t} = 0 \\ &\quad \text{otherwise,} \\ r_{2,t} &= 2B \quad \text{if } t \bmod 3 = 0 \quad \text{and} \quad r_{2,t} = 0 \\ &\quad \text{otherwise.} \end{aligned}$$

We claim that E can be partitioned as required if and only if there is a schedule for the constructed instance of $1|rm = 2, p_j = 1|L_{\max}$ with zero lateness.

First, assume that in the constructed instance of $1|rm = 2, p_j = 1|L_{\max}$ there is a schedule with zero lateness. Since there are $3m$ jobs of length 1, and the due date of each of the jobs equals $3m$, this implies that at each time unit we process one job. Let E_k consist three jobs $e_{k,1}$, $e_{k,2}$, and $e_{k,3}$, which are scheduled at time units $3(k-1)$, $3k-2$, and $3k-1$ where $1 \leq k \leq m$. Consider the first triple E_1 . By definition of a and the limited availability of the first raw material we have $\sum_{e \in E_1} s(e) = \sum_{e \in E_1} a_{1,e} \leq B$. Similarly it follows from the limited availability of the second raw materials that $3B - \sum_{e \in E_1} s(e) = \sum_{e \in E_1} (B - s(e)) = \sum_{e \in E_1} a_{2,e} \leq 2B$. Therefore, $\sum_{e \in E_1} s(e) \geq B$, and immediately $\sum_{e \in E_1} s(e) = B$. Continuing this argument for E_2, \dots, E_m , we observe that E_1, E_2, \dots, E_m is the required partition.

Now, assume that the partition exists. Clearly, scheduling triples E_1, E_2, \dots, E_m one-by-one without idle times we

get a feasible schedule with makespan $3m$ and hence zero lateness. \square

This proof can be easily adapted so as to apply to a C_{\max} objective function, instead of L_{\max} . This problem with C_{\max} objective function is addressed in subsection 4.2.

The remainder of this section is devoted to worst case analysis of approximation algorithms for $1|rm = 2, p_j = p|L_{\max}$. All the approximation algorithms under consideration will be based on the Earliest Due Date order. We say a schedule is *active* if, given the sequence in which the jobs are scheduled, the jobs are scheduled as early as possible, i.e., there is no unnecessary idle times. We call an approximation algorithm *active* if the algorithm outputs only active schedules. It is usual in approximation results for L_{\max} to assume that all due dates are negative. This implies that any solution has a positive objective function value. We adopt this convention as well. As in the case of a single raw material, we let $u_1 \leq u_2 \leq \dots \leq u_q$ be the moments in time at which a positive amount of raw material is supplied. Since we assume that the sum of the delivered quantities is exactly sufficient to complete all jobs, u_q is the earliest moment in time at which all required raw materials have arrived.

THEOREM 5: Any active approximation algorithm has a worst case ratio of at most 3.

PROOF: Consider the following three bounds on the maximum lateness:

1. $L_{\max} \geq p - d_1$, the minimum lateness of the first job.
2. $L_{\max} \geq u_q + p$, the earliest time moment at which the last job can be completed since all required raw materials are supplied.
3. $L_{\max} \geq np$, the earliest time moment by which all jobs can be processed.

Now, take an arbitrary active approximation algorithm and an arbitrary instance and consider the sequence in which the jobs are scheduled in the solution provided by the algorithm. At time u_q all required raw materials have arrived and hence an active schedule has to schedule the unfinished jobs without further delay. This takes no more than np time. Hence, the makespan of this schedule is at most $u_q + np$, and therefore, its lateness, is at most $u_q + np - d_1$.

But then $u_q + np - d_1 \leq 3 \max\{u_q + p, np, p - d_1\} \leq 3L_{\max}$ as required. \square

Consider the following EDD-based algorithms:

1. **Strict EDD:** Schedule the jobs in EDD order.

2. **Lazy EDD:** Wait until all raw materials have arrived, and schedule them in EDD order.
3. **Early EDD:** Starting at time moment $t = 0$, consider the set of jobs for which the required raw materials have arrived, and select from this set the one with earliest due date, set $t \rightarrow t + 1$ and continue.
4. **First Fit EDD:** Take the jobs in EDD order, and schedule the first one from the list at the earliest moment in time at which all required raw materials have arrived. Delete this job, and the latest possible supplied raw materials that it consumes from the instance and repeat. (The First Fit EDD algorithm was proposed to us by Thomas Erlebach).

The reader may note that the first three algorithms do not require any insight into future arrivals of raw materials supply. They are online algorithms in the sense that the scheduling decision regarding time t can be taken at time t , assuming all jobs are known, but without information regarding $r(s)$, $s > t$.

THEOREM 6: All four of the EDD based algorithms have a worst case ratio of 2.

PROOF: First of all consider the special strategy of Lazy EDD. It waits until u_q , the time moment at which all raw materials have arrived, and then schedules them without further idle time in EDD order. When compared to the other three EDD algorithms, we notice the following. Any of the other three algorithms may schedule some jobs before time u_q . Let W be the set of jobs scheduled before u_q . All jobs in $J \setminus W$ which have not been scheduled by u_q will be scheduled without further idle time in EDD order. But this implies that all jobs are scheduled no later than in the schedule constructed by Lazy EDD. Hence, if we prove that Lazy EDD has a worst case ratio of at most 2, this upper bound also applies to the other three algorithms. In fact, it applies to any algorithm that schedules the jobs not scheduled before u_q actively in EDD order.

Consider the relaxation of the problem in which the raw material availability is disregarded. For this relaxation the solution in which all jobs are scheduled without delay starting from time 0 in EDD order is optimal. Let l be the job maximizing $lp - d_l$. Then, the value of the optimal solution of this problem equals $lp - d_l$. Hence, $lp - d_l$ is a lower bound on the optimal solution value of the original problem as well. The solution provided by Lazy EDD schedules the jobs in the same EDD order. However, the first job is not scheduled at time 0 but rather at time u_q . Hence, the maximum lateness of Lazy EDD equals $u_q + lp - d_l$. Since u_q is a lower bound on L_{\max} , we deduce that

$$u_q + lp - d_l \leq 2 \max(u_q, lp - d_l) \leq 2L_{\max}. \quad (8)$$

Hence, to prove the theorem, it remains to show that the bound of 2 can be attained by all 4 EDD algorithms. This can be arranged using the following worst case instance:

1. All jobs have unit processing times.
2. There is one raw material: $n - 1$ units arrive at time 0, $n - 1$ units arrive at time $n - 1$.
3. Job 1: $d_1 = -2$, $a_1 = n - 1$.
4. Job $j \in J \setminus \{1\}$: $d_j = -1$, $a_j = 1$.

It can be easily verified that each of the EDD algorithms will start by processing job 1 at time 0. Since this job consumes all available raw material, the next job can only be scheduled when the next replenishment of raw materials arrives, at time $n - 1$. From there on, all remaining jobs can be processed without further delay, resulting in a lateness of $2n$. In the optimal solution, one simply schedules first all jobs $2, 3, \dots, n - 1$ consecutively starting from time zero, and then job 1, resulting in a lateness of $n + 2$. Hence the worst case ratio of each of the algorithms is $2n/(n + 2)$ which approaches 2 when n tends to infinity. \square

4. MAKESPAN MINIMIZATION

4.1. Dedicated Raw Materials ($1|ddc|C_{\max}$)

Consider the problem $1|ddc|C_{\max}$. Here again, the moments of supply of the dedicated raw materials for each of the jobs can be modeled by release dates. Clearly, for the problem $1|ddc|C_{\max}$ the following trivial algorithm provides an optimal solution: Schedule the jobs in order of nondecreasing release dates. Notice that this algorithm not only works for the case of all equal processing times, but for arbitrary processing times.

4.2. One Raw Material ($1|rm = 1|C_{\max}$)

For the problem $1|rm = 1, p_j = p|C_{\max}$ we introduce the following simple algorithm **A**: Schedule the jobs in nondecreasing order of raw material consumption a_j (breaking ties arbitrarily).

PROPOSITION 7: The solution obtained by algorithm **A** is an optimal solution for $1|rm = 1, p_j = p|C_{\max}$.

PROOF: Here we use a standard interchanging argument. Assume that there is an optimal solution where a job i is an immediate successor of a job j and $a_i \leq a_j$. If $a_i \leq a_j$, then interchanging the two jobs preserves feasibility with respect to inequalities (2) and (3). Therefore, it preserves the feasibility of the solution. Since all jobs have the same processing times, the interchange of jobs does not change the makespan. Repetitively applying the interchange argument,

we obtain an optimal solution in which all jobs are scheduled in order of nondecreasing a_j . \square

Now, we show that the extension of the problem to the case with arbitrary processing times is difficult.

THEOREM 8: The problem $1|rm = 1|C_{\max}$ is strongly NP-hard.

PROOF: Let us reduce the well-known (see, e.g., [5]) strongly NP-complete problem 3-PARTITION to $1|rm = 1|C_{\max}$. For notation on 3-PARTITION we refer to the previous section.

Define

$$\begin{aligned} J &= E, \\ a_e &= s(e), \quad e \in E, \\ p_e &= s(e), \quad e \in E, \\ r_t &= B \quad \text{if } t \bmod B = 0 \quad \text{and} \quad r_t = 0 \\ &\quad \text{otherwise.} \end{aligned}$$

We claim there is a required partition of set E if and only if there is a schedule for the constructed instance of $1|rm = 1|C_{\max}$ with makespan $C_{\max} = mB$.

Assume that in the constructed instance of $1|rm = 1|C_{\max}$ there is a schedule of length mB . Since the total workload of all jobs from E is mB and the schedule length is also mB we conclude that in the schedule there are no idle times. Consider the interval $[0, B]$. Since $B/4 < s(e) < B/2$ for any $e \in E$ and in the schedule there are no idle times, exactly three jobs $E_1 = \{e_{1,1}, e_{1,2}, e_{1,3}\}$ with total workload $\sum_{e \in E_1} p_e = \sum_{e \in E_1} s(e) \geq B$ are scheduled in the interval $[0, B]$. By definition of raw material consumptions, the feasibility of the schedule and inequality (3), we deduce that $\sum_{e \in E_1} a_e = \sum_{e \in E_1} s(e) \leq B$. Thus, $\sum_{e \in E_1} s(e) = B$. Applying the same arguments to intervals $[B, 2B]$, $[2B, 3B]$, \dots , $[(m-1)B, mB]$ we find that E_1, E_2, \dots, E_m is the required partition.

If there is a partition then it is trivial to construct a feasible schedule of length $C_{\max} = mB$: just schedule first all the jobs from E_1 , then schedule all the jobs from E_2 , and so on. Since at a time $3t$ there is enough raw material to schedule any triple E_k , $1 \leq k \leq m$ and the total workload on triple E_k is B , we get a feasible schedule without idle times. It implies that the makespan $C_{\max} = mB$. \square

We conclude this subsection by mentioning one more polynomially solvable case of $1|rm = 1|C_{\max}$. Namely, consider the problem with regular unit supply of raw material; i.e., at each time moment one additional unit of raw material becomes available.

Let us reduce the problem to the polynomially solvable flow shop problem $F2||C_{\max}$. For every instance $1|rm =$

$1|C_{\max}$ with regular unit supply of raw material, we construct a corresponding instance of $F2||C_{\max}$. The job set J will be unchanged, and every job $j \in J$ of the single machine problem will appear in the flow shop instance as follows. Its first operation $O_{1,j}$ corresponds to the collecting of the required raw materials. Hence, this operation takes time $p_{1,j} = a_j$. The second operation $O_{2,j}$ corresponds to the processing of job j on the machine, and hence takes time p_j . Clearly, the constructed flow shop correctly models the problem $1|rm = 1|C_{\max}$ with regular unit supply of raw material. Therefore, by Johnson's algorithm [8] an optimal solution for $F2||C_{\max}$ can be obtained in time $O(n \log n)$, and we claim the following proposition.

PROPOSITION 9: $1|rm = 1|C_{\max}$ with regular unit supply of raw material can be solved in time $O(n \log n)$ where n is the number of jobs.

4.3. Multiple Raw Materials ($1|rm|C_{\max}$)

Theorem 4 can be easily adjusted to demonstrate that $1|rm = 2, p_j = 1|C_{\max}$ is already strongly NP-hard. Hence, assuming $P \neq NP$, we cannot expect to find an exact polynomial time algorithm solving the problem to optimality. This section describes a straightforward approximation algorithm for $1|rm|C_{\max}$, with worst case ratio of 2.

Consider the following simple algorithm \mathcal{A}_1 . Starting from time moment u_q , the earliest moment in time at which all required raw materials have arrived, algorithm \mathcal{A}_1 actively schedules the jobs simply one-by-one in random order. Let $C_{\max}(\mathcal{A}_1)$ be the makespan of the solution obtained by this algorithm and C_{\max}^* be the optimal makespan. (Notice that Lazy EDD is a special case of \mathcal{A}_1 .) Then we have the following theorem.

THEOREM 10: $C_{\max}(\mathcal{A}_1)/C_{\max}^* \leq 2$ and the ratio of 2 is tight.

PROOF: Recalling that at time u_q the last required delivery of raw materials takes place, we derive that $C_{\max}^* \geq u_q + \min_{j \in J} p_j$. Further, $C_{\max}^* \geq \sum_{j \in J} p_j$. Since after time moment u_q the algorithm \mathcal{A}_1 schedules all jobs without idle times we have $C_{\max}(\mathcal{A}_1) = u_q + \sum_{j \in J} p_j \leq 2C_{\max}^*$.

It remains to show that the ratio of 2 is tight. Consider the case where we have only two jobs. Let the first job have processing time b and not require any raw material. The second job has unit processing time and requires one unit of raw material, which becomes available at time moment b . For this instance $C_{\max}^* = b + 1$ and $C_{\max}(\mathcal{A}_1) = 2b + 1$ which yields $\lim_{b \rightarrow \infty} C_{\max}(\mathcal{A}_1)/C_{\max}^* = \lim_{b \rightarrow \infty} (2b + 1)/(b + 1) = 2$. \square

Now consider the algorithm \mathcal{A}_2 (an extension of \mathcal{A}_1), which solves the problem with one raw material to optimal-

ity. \mathcal{A}_2 first orders the jobs in nondecreasing order of the sum of their raw material requirements. At any point in time it selects from this list the job with smallest index for which sufficient raw materials are available. It is not hard to verify that for any instance the makespan of the solution given by \mathcal{A}_2 cannot exceed the makespan of the solution given by \mathcal{A}_1 . The following instance shows that \mathcal{A}_2 has a worst case ratio of two nevertheless.

There are n jobs and two raw materials. Job 1 requires $n - 1$ units of raw material 1 and 0 units of raw material 2. Jobs $j, j = 2, \dots, n$ requires 1 unit of raw material 1, and $n + j - 3$ units of raw material 2. Notice that the index order yields the job list to be sorted in order of total raw material consumption. The raw material arrivals are as follows: At time 0, $n - 1$ units of raw material 1 arrive, and at time $n - 1$, another $n - 1$ units of raw material 1 arrive. At time $t, t = 1, \dots, n - 2, n + t - 3$ units of raw material 2 arrive. It is not hard to verify that the solution in which job $j, j = 2, \dots, n$ is scheduled at time $j - 1$, and job 1 is scheduled at time n is feasible, yielding a makespan of $n + 1$. On the other hand, \mathcal{A}_2 schedules job 1 at $t = 0$, and therefore cannot schedule any of the other jobs until $t = n - 1$, when the next quantity of raw material 1 arrives. It subsequently schedules jobs $j = 2, \dots, n$ at $j = n + i - 3$, yielding a makespan of $2n - 3$. Letting n approach infinity, a worst case ratio of two follows. Hence we have arrived at the following theorem.

THEOREM 11: $C_{\max}(\mathcal{A}_2)/C_{\max}^* \leq 2$ and the ratio of 2 is tight.

5. COMPUTATIONAL RESULTS

In this section we briefly report on computational results on the approximation algorithms for $1|rm|C_{\max}$ and for $1|rm|L_{\max}$. We have tested the algorithms in this paper on moderately sized instances for which we have been able to compute optimal solution values using ILOG CPLEX MIP Solver. For both problems we have generated two types of instances. First, a set of "random" instances, and second, a set of harder, "structured," instances, which were generated along the lines of the reduction of the hardness proof. Details of the computational study are available online [15].

We now first discuss the computational experiments for $1|rm|C_{\max}$. We have varied the number of jobs, from 10 to 30, raw materials, from 2 to 10, the maximum amount of a raw material required per job (maximum 40), and the maximum quantity of a raw material that arrives per time moment (maximum 40). For several choices of values for these quadruples, we have generated a number of otherwise random instances, altogether 140. \mathcal{A}_2 performs remarkably

well on these instances, it found the optimal solution on 137 of 140 instances. Apparently, they are not very hard to solve.

A second set of instances for $1|rm|C_{\max}$ is created along the lines of the reduction in the completeness proof. We generated 3-partition instances, and derived instances for $1|rm|C_{\max}$ from them, using the transformation of the completeness proof. The thus created instances are typically much harder to solve. \mathcal{A}_2 found the optimal solution on 6 out of 26 instances. However, it is not hard to find solutions which are close to optimal, and indeed \mathcal{A}_2 was always very close to optimal.

For $1|rm|L_{\max}$ we have created instances along similar lines as explained above for $1|rm|C_{\max}$. We did not do any experiments with Lazy EDD since it is a rather artificial algorithm, whose solutions are inferior to the others. On the random instances, First Fit EDD outperformed Strict EDD and Lazy EDD by a wide margin. In fact, it solved most of them to optimality, whereas the others did not. In addition, Strict EDD outperforms Early EDD on the random instances. In fact, the behavior of Early EDD is quite poor. Early EDD however performs slightly better than the other two on the hard instances which are constructed again using three partition instances.

6. CONCLUSIONS AND DISCUSSION

This paper opens up the area of scheduling problems with raw materials requirements, by analyzing several basic problems in this field. Raw materials requirements generalize the concept of release dates. However, our main motivation for studying scheduling problems with raw materials requirements is their increasing importance in Assemble To Order environments which are the dominant environment in today's leading industries.

Sections 3 and 4 show that several of the basic problems can be solved in polynomial time, and some online versions can even be solved to optimality, but the problems in which there are multiple shared raw materials are hard. However, the paper gives approximation algorithms with constant worst case ratios and together with the computational results of Section 5, they indicate that fast approximate solution methods that give satisfactory solutions for practical settings exist. The exact approximability status of the problems is however open. This is certainly worth further exploration. Likewise, the computational results indicate that exact solutions for small to moderate instances can be found using ILOG CPLEX, but improving exact solution methods is another area worthy of further research.

The basic single machine setting with unit or all equal processing times captures most of the essentials of an as-

sembly line, but is not general enough to cover all practical applications. Hence, models with arbitrary processing times can be studied more extensively, as is the case for problems with more than one machine. Further, in our analysis we have restricted the problem to model the assembly stage of a supply chain, and treat the customer orders as well as the raw material supplies as fixed. However, significant improvement in supply chain performance can be reached when scheduling of activities in consecutive stages is done collaboratively. Hence flow shop like models, where the objective is to optimize some overall supply chain performance measure are certainly important extensions of the models presented in this paper.

A less ambitious variation on this supply chain objective, is to consider a case where customer orders and raw material replenishment can be renegotiated, but at a certain cost. In this setting each customer order has a cost associated with being late, as can be modeled by weighted tardiness, and the raw material replenishment have costs associated with being early. This captures the process of avoiding being late in delivery through speeding up the supply, as it is frequently done in practice. The lateness minimization objective, or the completion time minimization objective are not fitting in this situation, and hence models with different objective functions are of interest as well.

Another important practical problem comes in sight then, namely the problem of due date quoting. State of the art ERP systems offer Available To Promise (ATP) and Capable To Promise (CTP) functionality. ATP refers to knowing how much of the end item stock, whether already available or known to become available, is not yet assigned to customer orders, and therefore available to be promised to new customers. CTP not only takes realized and planned production, but additionally also currently unoccupied production capacity into account. The question is then, how much are we capable to promise, not only by fulfilling from stock or planned production, but also by scheduling extra production. Hence, CTP functionality requires solving scheduling problems. One of these problems is the so-called due date quoting problem. In the due date quoting problem, the due date of a new customer order needs to be determined given a current production schedule in which all jobs are delivered on time. The question is to construct a new schedule which completes the new job as early as possible, under the restriction that all other jobs are still completed in time. This earliest moment is the due date that can be quoted to the new customer. This due date quoting problem pops up frequently in practice, but the tools to answer it correctly are lacking.

An extension of the due date quoting problem arises when the profit margins of the orders are known, as well as the costs for being late and the cost of early supply. Then the

problem of maximizing revenue for a given set of orders can be solved, and the problem of accepting new orders, or quoting due dates for them, can be stated in this same financial setting.

REFERENCES

- [1] N.J. Aquilano, R.B. Chase, and F.R. Jacobs, *Operations management for competitive advantage*, McGraw-Hill/Irwin, New York, 2001.
- [2] K.R. Baker, E.L. Lawler, J.K. Lenstra, and A.H.G. Rinnooy Kan, Preemptive scheduling of a single machine to minimize maximum cost subject to release dates and precedence constraints, *Oper Res* 31 (1983), 381–386.
- [3] J. Blazevicz, W. Cellary, R. Slowinski, and J. Weglarz, Scheduling under resource constraints—deterministic models, *Annals of Operations Research* 7, J.C. Baltzer AG, Basel, Switzerland, 1986.
- [4] P. Brucker, *Scheduling algorithms*, Springer, Berlin, 2001.
- [5] M.R. Garey and D.S. Johnson, *Computers and intractability: A guide to the theory of NP-completeness*, Freeman, San Francisco, 1979.
- [6] R.L. Graham, E.L. Lawler, J.K. Lenstra, and A.H.G. Rinnooy Kan, Optimization and approximation in deterministic sequencing and scheduling: A survey, *Ann Oper Res* 5 (1979), 287–326.
- [7] W.A. Horn, Some simple scheduling algorithms, *Naval Res Logist Quart* 21 (1974), 177–185.
- [8] S.M. Johnson, Optimal two- and three-stage production schedules with setup times included, *Naval Res Logist Quart* 1 (1954), 61–68.
- [9] J. Kallrath, Planning and scheduling in the process industry, *OR Spectrum* 24 (2002), 219–250.
- [10] W. Kubiak, G. Steiner, and J.S. Yeomans, Optimal level schedules for mixed-model multi-level just in time assembly systems, *Ann Oper Res* 57 (1995), 203–216.
- [11] P. Laborie, Algorithms for propagating resource constraints in AI planning and scheduling, *Artificial Intelligence* 143 (2003), 151–188.
- [12] J.K. Lenstra, A.H.G. Rinnooy Kan, and P. Brucker, Complexity of machine scheduling problems, *Ann Oper Res* 1 (1977), 343–362.
- [13] J. Miltenburg, Level schedules for mixed model assembly lines in just in time production systems, *Management Sci* 35 (1989), 192–207.
- [14] B.B. Simons, A fast algorithm for single processor scheduling, *Proc 19th IEEE Symp Found Comput Sci*, 1978, pp. 246–252.
- [15] University of Maastricht, Maastricht, The Netherlands, 2005, <http://www.personeel.unimaas.nl/a.grigoriev/rm/rm-experiments.pdf>